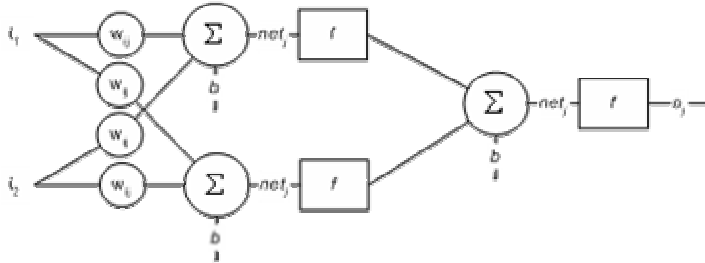


Arif Kusbandono
13299108
Tugas #1 AI

Algoritma Belajar XOR

Persoalan



Diinginkan agar jaringan di atas berfungsi seperti XOR. Dengan fungsi aktivasi signum, hubungan masukan keluaran yang diinginkan (*target*) :

i_1	i_2	t
0	0	-1
0	1	1
1	0	1
1	1	-1

Penurunan metoda backward-propagation

Secara umum, pemberian nilai baru bobot dan bias dari nilai mulanya pada proses belajar jaringan di atas menggunakan minimisasi fungsi *error* $E = \frac{1}{2}(t-o)^2$ terhadap bobot dan biasnya.

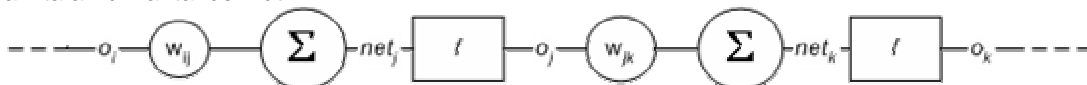
Fungsi aktivasi signum memberikan keluaran

$$o = f(x) = \begin{cases} 1 & , x > 0 \\ -1 & , x \leq 0 \end{cases}$$

maka kemungkinan yang terjadi adalah :

t	o	$\frac{1}{2}(t-o)^2$
-1	-1	0
-1	1	2
1	-1	2
1	1	0

Apabila dipetakan fungsi *error* E terhadap bobot akan sulit ditemukan jalan perubahan. Jika kita ambil rantai berikut :



dengan mengambil batas neuron i sebagai *input layer*, j sebagai *hidden layer*, dan k sebagai *output layer* akan dapat diturunkan rumus Δw maupun Δb yang akan digunakan dalam algoritma belajar jaringan tersebut.

Terdapat hubungan untuk *output layer*

$$net_k = \sum_j w_{jk} o_j$$

$$o_k = f(net_k)$$

dengan menggunakan fungsi error :

bobot yang diperlukan untuk mencapai error minimum ($\frac{\partial E}{\partial w}$) karena $f'(x) = 0$. Jadi selama belajar setiap kombinasi masukan hanya akan meloncat antara 2 (maksimum global E) dan 0 (minimum global E), dengan pergerakan diskret.

Jika dipakai fungsi aktivasi sigmoid (tidak setajam sign) yang terdefinisi turunannya, maka minimisasi error menggunakan turunan parsial $\frac{\partial E}{\partial w}$ dengan menggunakan aturan rantai dapat dirumuskan.

$$E = \frac{1}{2}(t_k - o_k)^2$$

maka untuk *output layer* :

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial net_k} \frac{\partial net_k}{\partial w_{jk}} = \frac{\partial (t_k - o_k)^2}{\partial o_k} \frac{\partial f(net_k)}{\partial net_k} \frac{\partial (\sum_j w_{jk} o_j)}{\partial w_{jk}}$$

$$\frac{\partial E}{\partial w_{jk}} = -(t_k - o_k) \frac{\partial f(net_k)}{\partial net_k} o_j = -o_j (t_k - o_k) f'(net_k)$$

Hubungan di atas merupakan *slope* kurva E terhadap salah satu bobot w_{jk} . Faktor $-\eta$ mengurangi bobot untuk mengecilkan error jika *slope* positif dan menambah bobot jika *slope* negatif, sebagaimana dinyatakan berikut:

$$w_{jk} \leftarrow w_{jk} + (-\eta)(-o_j (t_k - o_k) f'(net_k))$$

atau biasa dinyatakan dengan susunan:

$$w_{jk} \leftarrow w_{jk} + \Delta w_{jk}$$

$$\Delta w_{jk} = \eta \delta_k o_j$$

$$\delta_k = (t_k - o_k) f'(net_k)$$

Sedangkan untuk bias dengan aturan rantai yang sama dan $\frac{\partial net_k}{\partial b_k} = 1$ dinyatakan :

$$b_k \leftarrow b_k + \Delta b_k$$

$$\Delta b_k = \eta \delta_k$$

Pada perubahan bobot di atas saat proses belajar berlangsung, η menjadi faktor yang menentukan kecepatan proses (*learning rate*) dan kemungkinan error yang dicapai.

Untuk *hidden layer* :

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial net_k} \frac{\partial net_k}{\partial o_j} \frac{\partial o_j}{\partial net_j} \frac{\partial net_j}{\partial w_{ij}}$$

$$\frac{\partial E}{\partial w_{ij}} = -(t_k - o_k) \frac{\partial f(net_k)}{\partial net_k} w_{jk} \frac{\partial f(net_j)}{\partial net_j} o_i = -o_i w_{jk} (t_k - o_k) f'(net_k) f'(net_j)$$

$$w_{ij} \leftarrow w_{ij} + \Delta w_{ij}$$

$$\Delta w_{ij} = \eta \delta_j o_i$$

$$\delta_j = f'(net_j) \delta_k w_{jk}$$

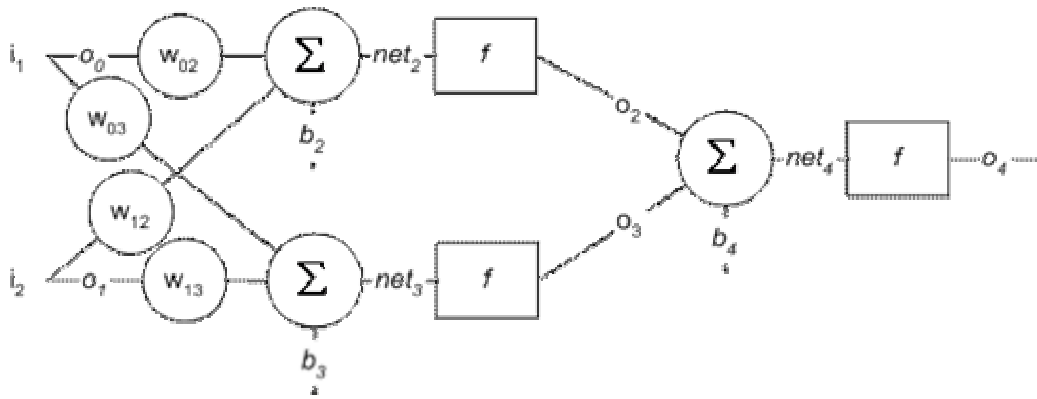
Untuk biasanya:

$$\frac{\partial E}{\partial b_j} = \frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial net_k} \frac{\partial net_k}{\partial o_j} \frac{\partial o_j}{\partial net_j} \frac{\partial net_j}{\partial b_j}$$

$$\frac{\partial E}{\partial b_j} = -(t_k - o_k) \frac{\partial f(net_k)}{\partial net_k} w_{jk} \frac{\partial f(net_j)}{\partial net_j} = -w_{jk} (t_k - o_k) f'(net_k) f'(net_j)$$

$$b_j \leftarrow b_j + \Delta b_j$$

$$\Delta b_j = \eta \delta_j$$



Dengan fungsi aktivasi $o_k = f(net_k) = \frac{1}{1 + e^{-net_k}}$ dan $f'(net_k) = o_k(1 - o_k)$ maka:

output layer:

$$\delta_4 = (t_4 - o_4) o_4 (1 - o_4)$$

$$\Delta b_4 = \eta \delta_4$$

hidden layer:

$$\delta_3 = o_3 (1 - o_3) \delta_4$$

$$\delta_2 = o_2 (1 - o_2) \delta_4$$

$$\Delta b_2 = \eta \delta_2$$

$$\Delta w_{02} = \eta \delta_2 o_0$$

$$\Delta w_{03} = \eta \delta_3 o_0$$

$$\Delta w_{12} = \eta \delta_2 o_1$$

$$\Delta w_{13} = \eta \delta_3 o_1$$

Target:

o_0	o_1	t_4
0	0	0
0	1	1
1	0	1
1	1	0

Implementasi pada program akan konsisten dengan indeks di atas, nilai target dirubah karena digunakan fungsi aktivasi yang petanya positif.

Hasil training : iterasi dibatasi 60000000, *learning rate* 0.6, E yang diinginkan tidak tercapai pada batas iterasi tersebut. Tampaknya arsitektur dengan w_{24} dan w_{34} sama dengan 1 tidak menguntungkan.

```
D:\main1_2.exe
Nilai inisial :
w02 = -8
w03 = 0.2
w12 = -3
w13 = 0.8
b4 = -3
b3 = -2
b2 = -2
0,0 = 0.43162
0,1 = 0.673647
1,0 = 0.431617
1,1 = 0.431619
(fungsi error) E = 0.806451
w02 = -7.99844
w03 = -22.5432
w12 = -2.5024
w13 = 23.1759
b4 = -0.275258
b3 = -12.0876
b2 = -12.0876
```